

Programmes de calcul

I D'un programme de calcul vers une expression

Voici un programme de calcul :

Programme P
 Choisir un nombre
 Ajouter 5
 Élever au carré
 Multiplier par 3
 Soustraire 12
 Écrire le résultat

- 1) Montrer que pour le nombre 1 au départ, on obtient 96.
- 2) Calculer le résultat pour le nombre -2 .
- 3) Recommencer pour le nombre 0.
- 4) Pour un nombre quelconque x au départ, écrire l'ensemble des calculs de ce programme en une expression que l'on nommera $P(x)$.

$P(x) = \dots$

- 5) En Python, ce programme s'écrit dans un **éditeur Python** comme ci-contre.

La ligne 2 est à compléter avec l'expression de $P(x)$.

L'écrire et l'enregistrer dans :

Documents\travail 2nde\SNT\pgmcalcul1.py

On l'utilise ensuite **dans la console** comme ci-contre.

Vérifier pour $P(-2)$ et $P(0)$.

```
pgmcalcul1.py x
1 def P(x):
2     return ...|
```

```
En cours d'exécution: pgmcalcul1.py
>>> P(1)
96
>>> |
```

II D'une expression vers un programme de calcul.

- 1) Écrire les expressions programmées ci-contre.

$Q(x) = \dots$

$R(x) = \dots$

$S(x) = \dots$

- 2) Écrire le programme de calcul correspondant à Q, R et T.

```
4 def Q(x):
5     return 3*(x**2-5)+12
6
7 def R(x):
8     return (5*x-3)**2+12
9
10 def S(x):
11     return (5*(x+3)-12)**2|
```

Programme Q Choisir un nombre	Programme R Choisir un nombre	Programme S Choisir un nombre

- 3) Appliquer ces programmes de calcul aux nombres 1 ; -2 et 0 et noter les résultats

$Q(1) =$

$R(1) =$

$S(1) =$

$Q(-2) =$

$R(-2) =$

$S(-2) =$

$Q(0) =$

$R(0) =$

$S(0) =$

- 4) Saisir les expressions dans l'éditeur Python puis contrôler ensuite les résultats avec la console.

III Obtenir plusieurs valeurs du programme de calcul.

Voici deux méthodes pour obtenir les valeurs du programme de calcul P pour les nombres entiers de -3 à 3.

1) Avec une boucle bornée

Le programme calcule et affiche P(i) pour i prenant les valeurs de -3 à 3.

```
13 for i in [-3, -2, -1, 0, 1, 2, 3]:
14     print(i, "donne", P(i))
```

Tester ce script.

2) Avec une boucle conditionnelle

Le programme part de i = -3.

Tant que i ≤ 3, il calcule et affiche P(i) puis, il ajoute 1 à i et recommence.

```
16 print("-" *16)
17
18 i = -3
19 while i <= 3:
20     print(i, "donne", P(i))
21     i = i+1
```

Tester ce script.

IV Étude de deux autres programmes de calcul

Programme A
 Choisir un nombre
 Ajouter 7
 Calculer l'inverse du nombre obtenu
 Multiplier par 2
 Soustraire 3
 Écrire le résultat

Programme B
 Choisir un nombre
 Calculer d'une part, la somme de 19 et du triple de ce nombre
 Calculer d'autre part la somme de 7 et du nombre de départ
 Calculer l'opposé du quotient des deux résultats obtenus
 Écrire le résultat

- Appliquer ces programmes de calcul aux nombres 3 et -2 et noter les résultats
 $A(3)=$ $A(-2)=$ $B(3)=$ $B(-2)=$
- Pour un nombre quelconque x au départ, écrire l'ensemble des calculs de chacun de ces programmes en une expression que l'on nommera respectivement A(x) et B(x).

$A(x)=...$ $B(x)=...$

- Compléter le script Python ci-contre avec les deux expressions obtenues.

L'écrire et l'enregistrer dans :

Documents\travail 2nde\SNT\pgmcalcul2.py

```
pgmcalcul2.py X
1 def A(x):
2     return ...
3
4 def B(x):
5     return ...
```

- Contrôler les résultats précédents avec la console.
- Compléter le script pour qu'il affiche les valeurs du premier programme pour les nombres entiers de -2 à 5 à l'aide d'une boucle bornée.
- Compléter le script pour qu'il affiche les valeurs du deuxième programme pour les nombres entiers de -2 à 5 à l'aide d'une boucle conditionnelle.
- Faire fonctionner le script. Que peut-on remarquer ?
 Écrire la réponse sous forme d'un commentaire à la fin du script.
 ☞ Dans un script, un commentaire commence par le symbole #.
- Déposer le **lien actif** et/ou le **fichier programme** dans **Devoir 4 Programmes de calcul** dans **elyco**.